Review of Statistics

In [ ]:
```
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np
from numpy import random
import pandas as pd
%pylab inline
```

Functions that help with plotting

In [ ]:
```
def prob(rv, a, b):
    return 1-(rv.cdf(a)+(1-rv.cdf(b)))

def plotDist(x, func, title, l, xlabel, ylabel) :
    #plt.figure(fig)
    plt.plot(x, func, 'bo', ms=4, label=l) # plot func for elements of x
    xl = plt.gca().get_xlim()
    plt.hlines(0, xl[0], xl[1], linestyles='--', colors='#999999') #lines on Y-axi
s
    plt.gca().set_xlim(xl)
    plt.vlines(x, 0, func, colors='r', lw=2, alpha=0.5) # lines on X-axis
    plt.legend(loc='best', frameon=False)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
    plt.show()

def plotDist2(x, func, title, l, xlabel, ylabel) :
    plt.plot(x, func, 'b-', lw=2, alpha=0.6, label=l) # plot func for elements of
x
    xl = plt.gca().get_xlim()
    plt.hlines(0, xl[0], xl[1], linestyles='--', colors='#999999') #lines on Y-axi
s
    plt.gca().set_xlim(xl)
    #plt.vlines(x, 0, func, colors='r', lw=2, alpha=0.5) # lines on X-axis
    plt.legend(loc='best', frameon=False)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)

def plotHistDist(func, x, r, title, l, xlabel, ylabel):
    plt.hist(r, normed=True, histtype='stepfilled', alpha=0.2)
    plotDist2(x, func, title, l, xlabel, ylabel)
```

Basic Concepts

Random Sampling

Statistical experiments involve observations of a sample selected from a larger body of data, existing or conceptual, called the population

- Size of population N is much larger than size of sample n
- A sample can contain more than one item and can be measured for one or more random variable
- A sample can be drawn with or without replacement
- A simple random sample is drawn in such a way that every possible sample has an equal probability of being selected

Example: Given $N = 10$ and $n = 2$. There are

$$\binom{10}{2} = \frac{10 \cdot 9}{1 \cdot 2} = 45$$

possible combinations of two items (samples). A simple random sampling will not biased towards any of the samples.

Statistics

We may know that the probability distribution for a large population has a certain type of distribution function with unknown parameters θ

To estimate θ, we take a random sample of size n and treat the values in the sample as an observation of n random variables $Y_1, Y_2, \ldots, Y_n$.

If n is small enough compared to population size N, $Y_1, Y_2, \ldots, Y_n$ can be assumed to be independent and identically distributed (iid) random variables.

A statistic $\hat{\theta}$ is a function of random variables in the sample

- which is also a random variable
- used to estimate or reason about θ
- has a histogram through repeated sampling (i.e., do the sampling many times)
- A theoretical model of the histogram results in a sampling distribution for $\hat{\theta}$ and can be used to learn properties of θ

Sampling Distribution

Assume that a population has a normal distribution with unknown mean μ and variance σ^2. With a random sample $Y_1, Y_2, \dots, Y_n$ taken from the population, we can compute the following statistics.

- Sample mean:

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

- Sample variance

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

Statistics $\bar{Y}$ and S^2 are functions of random variables. They are themselves random variables, too. From one sample, we can compute a particular value for $\bar{Y}$ and S^2. By repeated sampling, we can observe other values, too. So, they have sampling distributions. Specifically,

- $\bar{Y}$ has a normal distribution with $E(\bar{Y}) = \mu_{\bar{Y}} = \mu$ and $V(\bar{Y}) = \sigma^2_{\bar{Y}} = \sigma^2/n$
- For S^2, we know that

$$\frac{(n-1)S^2}{\sigma^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

  has a χ^2 distribution with $(n-1)$ degree of freedom

Example

In this example, we assume that the population has a normal distribution. We then repeatedly take random samples from the population, compute sample means and sample variances, and compare the histogram of sample means with the corresponding normal distribution, and compare the histogram of sample variances with the corresponding chi-squared distribution.

In [ ]:
```
mu, sigma = 5, 2.1
p = stats.norm(mu, sigma) # Population distribution
k = 50  # number of times of re-sampling
sk = 10 # sample size
sm = np.zeros(k)
sv = np.zeros(k)

# Repeat the sampling k times
for i in range(k):
    Y = p.rvs(size=sk) # take a sample
    sm[i] = Y.mean()   # find sample mean
    sv[i] = Y.var()    # find sample variance

# compare with normal distribution
sigma2 = sigma/np.sqrt(sk)
rv = stats.norm(mu, sigma2)
x = np.linspace(rv.ppf(0.001), rv.ppf(0.999), 50)
label = "loc={}, scale={}".format(mu, sigma2)
plotHistDist(rv.pdf(x), x, sm, 'normal pdf', label, 'value of rv', 'probability')
plt.show()
```

In [ ]:
```
# Compare with chi-squared distribution
svv = ((sk-1)*sv) / sigma**2
rv2 = stats.chi2(sk-1)
x = np.linspace(rv2.ppf(0.001), rv2.ppf(0.999), 50)
label = "loc={}, scale={}".format(mu, sk-1)
plotHistDist(rv2.pdf(x), x, svv, 'chi2 pdf', label, 'value of rv', 'probability')
plt.show()
```

Alternatively, we can treat a data set as a set of samples randomly draw from a underlying unknown population, and compare the sample's histogram with some known probability distributions.

Sampling Distributions Involving Z

Let $Y_1, Y_2, \ldots, Y_n$ be a random sample from (a population with) a normal distribution with (unknown) mean μ and variance σ^2. Then

$$Z_i = \frac{Y_i - \mu}{\sigma}$$

for $1 \leq i \leq n$, are independent standard normal variables (that is, $\mu = 0$ and $\sigma = 1$) and

$$\sum_{i=1}^{n} Z_i^2 = \sum_{i=1}^{n} (\frac{Y_i - \mu}{\sigma})^2$$

has a χ^2 distribution with n degree of freedom

Sampling Distribution of T and F

If Z is a standard normal random variable, W is a χ^2 distributed random variable, and Z and W are independent, then

$$T = \frac{Z}{\sqrt{W/v}}$$

has a student's t-distribution with v degree of freedom

If W_1 and W_2 are independent χ^2 distributed random variables with v_1 and v_2 degree of freedom, respectively, then

$$F = \frac{W_1/v_1}{W_2/v_2}$$

has an F distribution with v_1 numerator degree of freedom and v_2 denominator degree of freedom

Central Limit Theorem

Let $Y_1, Y_2, \ldots, Y_n$ be independent and identically distributed random variables with $E(Y_i) = \mu$ and variance $V(Y_i) = \sigma^2 < \infty$. Distribution function of

$$U = \frac{\bar{Y} - \mu}{\sigma/\sqrt{n}} = \sqrt{n}(\frac{\bar{Y} - \mu}{\sigma})$$

converges to a standard normal distribution as $n \to \infty$

- For a large enough sample, say $n > 35$, U can be assumed to have a standard normal distribution
- This is true for any type of population distribution (normal or not)

Using Central Limit Theorem: An Example

Assume that Achievement test scores of all high schools in a state have a mean of 60 and variance of 64. If a random sample of 100 students from a large high school has a sample mean test score of 58, is there an evidence that this high school performs poorly?

Solution

From the given info, $n = 100, \mu = 60, \sigma^2 = 64, \bar{Y} = 58$, we wan to estimate $P(\bar{Y} \leq 58)$

By Central Limit Theorem, $U = \sqrt{100}(\bar{Y} - 60)/\sqrt{64}$ has approximately a standard normal distribution. So

$$P(\bar{Y} \leq 58) \simeq P(U \leq \frac{\sqrt{100}(58 - 60)}{\sqrt{64}}) = P(U \leq -2.5) = 0.0062$$

Therefore in this population, it is very unlikely for $\bar{Y} \leq 58$. Thus, this high school really performed poorly.

The calculation can be performed in Python as follows.

In [ ]:
```
n, mu, var, Ybar = 100, 60, 64, 58
U = sqrt(n)*(Ybar-mu)/ sqrt(var)
rv = stats.norm()
rv.cdf(U)
```

Distribution Estimation

Suppose we know or assume that in a population, one or more random variable has a certain probability distribution $F(\theta)$, where θ is a set of parameters, e.g., μ and σ^2, with unknown values. How can we find an estimate of these parameters?

We take a random sample $Y_1, Y_2, \ldots, Y_n$ of a large enough size n and use it to estimate the value θ. The estimation of the parameters may not be exact and may not be accurate.

Estimators, Bias and Mean Square Error

An estimator $\hat{\theta}$ is a formula that calculates a value of a population parameter θ based on a sample

- $\hat{\theta}$ is unbiased if $E(\hat{\theta}) = \theta$, biased otherwise
- The mean square error of the estimator is
$$MSE(\hat{\theta}) = E((\hat{\theta} - \theta)^2) = V(\hat{\theta}) - (B(\hat{\theta}))^2$$
where $B(\hat{\theta}) = E(\hat{\theta}) - \theta$ is the bias.

Given a random sample $Y_1, Y_2, \ldots, Y_n$ taken from a population with mean μ and variance σ^2.

- Sample mean $\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i$ is an unbiased estimator for population mean μ, i.e., $E(\bar{Y}) = \mu$
- Sample variance $S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (Y_i - \bar{Y})^2$ is an unbiased estimator for population variance σ^2, i.e., $E(S^2) = \sigma^2$
- But, $S'^2 = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \bar{Y})^2$ is not an unbiased estimator for population variance, because
$$E(S'^2) = \frac{\text{n-1}}{n}\sigma^2 \neq \sigma^2$$

Example

Repeatedly taking random samples from a normal destributed population, calculate sample means and sample variances, varify that these statistics are unbiased.

In [ ]:
```
mu, sigma = 5, 2.15 # population mean and standard deviation

repeat = 100
n = 50
mus = np.zeros(repeat)
S2s = np.zeros(repeat)

# take samples
for i in range(repeat) :
    s = pd.Series(np.random.normal(mu, sigma, n))
    mus[i] = s.mean()
    S2s[i] = stats.tstd(s)**2

print("E(YBar) = ", mus.mean(), " mu = ", mu)
print("E(S^2) = ", S2s.mean(), " sigma^2 = ", sigma**2)

rv = stats.norm(mu, sigma)
x = np.linspace(rv.ppf(0.001), rv.ppf(0.999), 50)
label = "loc={}, scale={}".format(mu, sigma)
plotHistDist(rv.pdf(x), x, S2s, 'normal pdf', label, 'value of rv', 'probability')
plt.show()
```

Confidence Intervals

With repeated sampling, an estimator $\hat{\theta}$ becomes a random variable with a sampling distribution. We can then compute the probability that $\hat{\theta}$ falls into a given range of values

For a given significance (or test) level $0 < \alpha < 1$, the estimator $\hat{\theta}$ has a $1 - \alpha$ confidence interval as follows.

- $[\hat{\theta}_L, \hat{\theta}_H]$ if $P(\hat{\theta}_L \leq \theta \leq \hat{\theta}_H) = 1 - \alpha$ (called two-sided)
- $[\hat{\theta}_L, \infty)$ if $P(\hat{\theta}_L \leq \theta) = 1 - \alpha$ (one-sided)
- $(-\infty, \hat{\theta}_H)$ if $P(\theta \leq \hat{\theta}_H) = 1 - \alpha$ (one-sided)

where $1 - \alpha$ is the confidence coefficient, often set to .95, .99, etc., and $\hat{\theta}_L, \hat{\theta}_H$ are boundaries of the interval

Confidence Interval: An Example

Assume that Y is a single observation sample from an exponential distribution with density function

$$f_Y(y) = \begin{cases} \frac{1}{\theta} e^{-y/\theta}, & 0 \leq y \\ 0, & otherwise \end{cases}$$

find the confidence interval of θ with confidence coefficient $1 - \alpha = 0.9$

Solution

- Let $U = Y/\theta$, then $f_U(u) = e^{-u}$, for $u \geq 0$
- Want to find a and b s.t., $P(a \leq U \leq b) = 0.9$
- Let $P(U < a) = 1 - e^{-a} = 0.05$ and $P(U > b) = e^{-b} = 0.05$, so, $a = 0.051, b = 2.996$

Then from $P(0.051 \leq \frac{Y}{\theta} \leq 2.996) = 0.9$, we get $P(\frac{Y}{2.996} \leq \theta \leq \frac{Y}{0.051}) = 0.9$

Confidence Interval: Another Example

A supermarket has a large population of customers. The number of minutes a randomly selected customer spent in shopping at this supermarket is a random variable of which the probability distribution has an unknown mean $\theta = \mu$. To estimate the range of values for μ, we observed $n = 64$ randomly selected customers at the supermarket and found that their mean shopping time is $\bar{Y} = 33$ minutes with a variance of $S^2 = 256$. We want to find $1 - \alpha = .90$ confidence interval of μ, the mean shopping time of the population.

Solution

Use $\bar{Y}$ as the estimator for μ, i.e., let $\hat{\theta} = \bar{Y}$. We know for large samples, $Z = \frac{\hat{\theta}-\theta}{\sigma_{\hat{\theta}}}$ has a standard normal distribution, and $1 - \alpha$ confidence interval for Z is $[-z_{\alpha/2}, z_{\alpha/2}]$

$$1 - \alpha = P(-z_{\alpha/2} \leq \frac{\hat{\theta} - \theta}{\sigma_{\hat{\theta}}} \leq z_{\alpha/2}) = P(\hat{\theta} - z_{\alpha/2}\sigma_{\hat{\theta}} \leq \theta \leq \hat{\theta} + z_{\alpha/2}\sigma_{\hat{\theta}})$$

$\sigma_{\hat{\theta}} = \sqrt{V(\hat{\theta})}$, standard deviation of $\hat{\theta}$, is also called standard error

In this example, $\hat{\theta} = \bar{Y}$ and $\sigma_{\bar{Y}} = \frac{\sigma}{\sqrt{n}}$. Since σ^2 is unknown, we use S^2 as its estimate, so

$$Z = \frac{\bar{Y} - \mu}{S/\sqrt{n}}$$

which has a standard normal distribution and the $1 - \alpha$ confidence interval for Z is

$$-z_{\alpha/2} \leq \frac{\bar{Y} - \mu}{S/\sqrt{n}} \leq z_{\alpha/2}$$

or equivalently,

$$\bar{Y} - z_{\alpha/2}(\frac{S}{\sqrt{n}}) \leq \mu \leq \bar{Y} + z_{\alpha/2}(\frac{S}{\sqrt{n}})$$

A solution using Python is given below.

In [ ]:
```
# From the sample
n = 64
YBar = 33
S = np.sqrt(256) # as estimator of sigma

# Test level
alpha = 1 - 0.75
z_half_alpha = stats.norm().ppf(1-alpha/2)

# Find confidence interval for mu
l = YBar - z_half_alpha*(S/np.sqrt(n))
r = YBar + z_half_alpha*(S/np.sqrt(n))
s = "The confidance interval of mu with probability of {0:3.2f} is [{1:4.2f}, {2:4
.2f}]"
P = 1-alpha
print(s.format(P, l, r))
```

Large vs Small Samples

For large samples, a number of unbiased estimators will have a normal distribution.

- $\bar{Y}$ has a normal distribution with $E(\bar{Y}) = \mu$, $\sigma_{\bar{Y}} = \frac{\sigma}{\sqrt{n}}$
- $\hat{p} = \frac{Y}{n}$ has a normal distribution with $E(\hat{p}) = p$, $\sigma_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$
- Also, $\bar{Y}_1 - \bar{Y}_2$ and $\hat{p}_1 - \hat{p}_2$ have normal distributions
- We can also assume Z has standard normal distribution

For small samples, Z no longer has a standard normal distribution. However, if population has a normal distribution, we may be able to use more complex functions that has a t-distribution.

- $T = \frac{\bar{Y}-\mu}{S/\sqrt{n}}$ has a t-distribution with $n-1$ degree of freedom

Maximum Likelihood Estimate

Let $Y_1, Y_2, \ldots, Y_n$ be a random sample from a normal distribution with unknown μ and σ^2. The following steps can find the Maximum Likelihood Estimate (MLE) of μ and σ^2

- define likelihood:

$$L(\mu, \sigma^2) = f(y_1, y_2, \ldots, y_n|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(y_1-\mu)^2}{2\sigma^2}) \cdots \frac{1}{\sigma\sqrt{2\pi}} exp(\frac{-(y_n-\mu)^2}{2\sigma^2})$$
$$= (\frac{1}{2\pi\sigma^2})^{n/2} exp(\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2)$$

- obtain Log-likelihood:

$$\ln[L(\mu, \sigma^2)] = -\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln 2\pi - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2$$

- Set partial derivatives to zeros

$$\frac{\partial\{\ln[L(\mu, \sigma^2]\}}{\partial\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\mu) = 0$$
$$\frac{\partial\{\ln[L(\mu, \sigma^2]\}}{\partial\sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(y_i-\mu)^2 = 0$$

- Solve for μ and σ^2 to obtain

$$\hat{\mu} = \bar{Y}$$
$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{Y})^2$$

  These estimates are calculated from the sample.

Elements of Statistical Test

A statistical test uses a random sample to test some hypothesis about statistics of a population.

To perform a statistical test, we need to

- Make two hypothesis:
  - H_0 null hypothesis
  - H_a alternative hypothesis.

  We want to reject H_0 and therefore prove H_a
- Design an experiment and take a random sample
- Compute statistics from the sample
- Determine a rejection region (RR), we can reject H_0 in favor of H_a only if the statistic falls into the rejection region

Two types of errors of the test

- Type I error occurs when H_0 is rejected by mistake. The probability of Type I error is α, the level of the test.
- Type II error occurs when H_0 is accepted by mistake. The probability of Type II is β

Large Sample Z-Test

Let θ be a parameter of a population and θ_0 be a particular value of θ (a threshold value). We want to determine whether $\theta > \theta_0$ (one-tailed, upper test) or $\theta \neq \theta_0$ (two-tailed test).

Test setup:

1. Take a large random sample from the population
2. Set hypothesis:
   - $H_0 : \theta = \theta_0$
   - $H_a : \theta > \theta_0$ (one-tailed, upper) or $H_a : \theta \neq \theta_0$ (two-tailed)
3. Choose an estimator $\hat{\theta}$ for θ
   - Use sample mean $\bar{Y}$ to estimate population mean μ
   - Use sample proportion $\hat{p}$ to estimate population proportion p
4. Compute the statistic $Z = \frac{\hat{\theta}-\theta_0}{\sigma_{\hat{\theta}}}$, which has a standard normal distribution for large samples
5. Find the reject region for a given test level α under the assumption that $H_0 : \theta = \theta_0$ is true.
   - For one-tailed upper test, the RR is (z_α, ∞), where $\alpha = P(Z > z_\alpha)$
   - For two-tailed test, the RR is $(-\infty, -z_{\alpha/2})$ and $(z_{\alpha/2}, \infty)$
6. Determine the test outcome
   - For one-tailed upper test, if $Z = z > z_\alpha$, we reject H_0 and conclude that $\theta > \theta_0$
   - For two-tailed test, if $|Z| > z_{alpha/2}$ (or equivalently, $Z < -z_{alpha/2}$ or $Z > z_{alpha/2}$), we reject H_0 and conclude that $\theta \neq \theta_0$

Large Sample One-Tailed (upper) Z-Test Example

A machine in a factory will be replaced if it produces more than 10% defectives among a large lot of items produced in a day. If 15 defectives are found in a random sample of 100 items, is it enough evidence that the machine should be replaced? Use test level 0.01

Solution

We need to determine if the proportion of defectives in the population is more than 10%. This is a one-tailed upper test.

- Let Y be the number of defectives in a sample, p be the proportion of defective in population. Sample size $n = 100$, threshold $p_0 = 0.1$, and test level $\alpha = 0.01$.
- Let hypothesis be $H_0 : p = .10$ and $H_a : p > .10$
- Use $\hat{p} = \frac{Y}{n}$ as the estimator, which is unbiased. Assume that H_0 is true, we can estimate $\sigma_{\hat{p}} = \sqrt{\frac{p_0(1-p_0)}{n}}$
- Calculate

$$Z = \frac{\hat{p} - p_0}{\sigma_{\hat{p}}}$$

- Compare Z with z_α

In [ ]:
```
# From the sample
Y = 15
n =100
p0 = 0.10
pHat = Y/n # the estimator
sigma = np.sqrt((p0*(1-p0))/n) # estimated assuming H_0 holds
mu = p0

# compute Z
z = (pHat-p0)/sigma

# Find z_{alpha}
alpha = 0.01  # test level
z_alpha = stats.norm().ppf(1-alpha)

# Perform large sample test
if (z>z_alpha) :
    print("Since Z={0:4.2f} > Z_alpha={1:4.2f}, we reject H_0".format(z, z_alpha))
else:
    print("Since Z={0:4.2f} <= Z_alpha={1:4.2f}, we cannot reject H_0".format(z, z
_alpha))
```

Large Sample Two-Tailed Z-Test Example

We know that the mean of a measurement for special type of object is 8.5. Given the following sample, we want to determine if the population mean equals to 8.5 with a test level of 0.01.

Solution

This is a two-tailed test. Let the hypotheses be: H_0: $\mu = 8.5$, H_a: $\mu \neq 8.5$.

In [ ]:
```
# From the sample
sample = np.array([10.73, 8.89, 9.07, 9.20, 10.33, 9.98, 9.84, 9.59,
                   8.48, 8.71, 9.57, 9.29, 9.94, 8.07, 8.37, 6.85,
                   8.52, 8.87, 6.23, 9.41, 6.66, 9.35, 8.86, 9.93,
                   8.91, 11.77, 10.48, 10.39, 9.39, 9.17, 9.89, 8.17,
                   8.93, 8.80, 10.02, 8.38, 11.67, 8.30, 9.17, 12.00, 9.38])
n = sample.size
YBar = sample.mean()
S = stats.tstd(sample)
print(n, YBar, S)

# Compute Z, assuming \mu = 8.5, \sigma_YBar = A/sqrt(n)
mu_0 = 8.5
Z = (YBar - mu_0)/(S/np.sqrt(n))

# Find z_{alpha/2}
alpha = 0.03
half_alpha = alpha/2
z_half_alpha = stats.norm().ppf(1-half_alpha)

# Large Sample test
if (np.abs(Z)>z_half_alpha) :
    print("Since |Z|={0:4.2f} > Z_half_alpha={1:4.2f}, we must reject H_0".format(
        np.abs(Z), z_half_alpha))
else:
    print("Since |Z|={0:4.2f} <= Z_half_alpha={1:4.2f}, we cannot reject H_0".form
at(
        np.abs(Z), z_half_alpha))
```

p-Value: Observed (or Attained) Significance

The p-value is the probability of observing a value of the test statistic computed under the assumption that H_0 is true. At any test level α that is greater than the p-value, we will reject H_0 and for any α that is less than the p-value, we do not have sufficient evidance to reject H_0.

In [ ]:
```
# For current example
Z = stats.norm()
print("The observed Z is z_0={0:5.4f}".format(z))
print("P(Z<=-{0:5.4f})={1:4.3f} and P(Z>={0:5.4f})={2:4.3f}".format(
       z, Z.cdf(-z), (1-Z.cdf(z))))
p = Z.cdf(-z)*2
print("p-value = P(Z<=-{0:5.4f}) + P(Z>={0:5.4f}) = {1:4.3f}".format(z, p))
print("For any alpha less than {0:4.3f}, we cannot reject H_0".format(p))
print("For any alpha greater than {0:4.3f}, we must reject H_0".format(p))
```

In general, for a given sample, statistic W, a threshold w_0 from the sample, and null hypothesis $H_0 : W = w_0$, the statistical test can also be performed as follows.

- if we want to prove $W \geq w_0$, compute $p\text{-value} = P(W \geq w_0)$ assuming H_0 is true
- if we want to prove $W \leq w_0$, compute $p\text{-value} = P(W \leq w_0)$ assuming H_0 is true
- If we want to prove $w_1 \leq W \leq w_0$, compute $p\text{-value} = P(W \leq w_0) + P(W \geq w_0)$ assuming H_0 is true
- For any α that is larger than the p-value, the null hypothesis H_0 should be rejected

Linear Statistical Model

Let Y be a response variable, $x_1, x_2, \ldots, x_k$ be independent variables (not to be confused with probabilistic independence). The linear regression model of Y is

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \epsilon$$

where ϵ is a random error with $E(\epsilon) = 0$, and β_i's are unknown parameters. So

$$E(Y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

- x_i can be any function of other variables meaningful to an application
- If only β_0, β_1 exist, the model is simple linear, otherwise, multi-linear
- The task is to estimate β_i

Method of Least Squares

Consider a population with data points (X, Y), and assume a simple linear model $E(Y) = \beta_0 + \beta_1 X$

- Given a set of data points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Estimate β_0, β_1
- If we can find some $\hat{\beta}_0, \hat{\beta}_1$ and use the model to estimate

  $$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

  the error will be $y_i - \hat{y}_i$ and sum of square of error is

  $$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

- We want to find $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize SSE. To do so, Set partial derivatives of SSE to zero and solve for $\hat{\beta}_0$ and $\hat{\beta}_1$

  $$\frac{\partial SSE}{\partial \hat{\beta}_0} = -2\left(\sum_{i=1}^{n} y_i - n\hat{\beta}_0 - \hat{\beta}_1 \sum_{i=1}^{n} x_i\right) = 0$$
  $$\frac{\partial SSE}{\partial \hat{\beta}_1} = -2\left(\sum_{i=1}^{n} x_i y_i - \hat{\beta}_0 \sum_{i=1}^{n} x_i - \hat{\beta}_1 \sum_{i=1}^{n} x_i^2\right) = 0$$

- Solution:

  $$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$
  $$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

  where

  $$S_{xy} = \sum_{i=1}^{n} (x_1 - \bar{x})(y_i - \bar{y})$$
  $$S_{xx} = \sum_{i=1}^{n} (x_1 - \bar{x})^2$$

Method of Least Squares: Example

Given X ={-2, -1, 0, 1, 2} and Y ={0, 0, 1, 1, 3}. Find $\hat{\beta}_1$, $\hat{\beta}_0$ for a linear model $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

In [ ]:
```
x = np.array([-2, -1, 0, 1, 2])
y = np.array([0, 0, 1, 1, 3])
xBar = x.mean()
yBar = y.mean()
Sxy = ((x-xBar)*(y-YBar)).sum()
Sxx = ((x-xBar)**2).sum()
B_1 = Sxy/Sxx
B_0 = yBar-B_1*xBar
print("YHat = ", B_0, " + ", B_1, "x" )
```

Method of System Equation

For each pair of (x, y), we have an equation

$$\hat{\beta}_0 \cdot 1 + \hat{\beta}_1 \cdot x = y$$

So, the problem can be modeled as a system of equations, which has the matrix equation

$$X\hat{B} = Y$$

or equivalently,

$$(X'X)\hat{B} = X'Y$$

Solution

$$\hat{B} = (X'X)^{-1}X'Y$$

Method of System Equation: Example

Given X ={-2, -1, 0, 1, 2} and Y ={0, 0, 1, 1, 3}. Find $\hat{\beta}_1$, $\hat{\beta}_0$ for a linear model $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

In [ ]:
```
# Using matrix operations
X =np.array([[1, -2], [1, -1], [1, 0], [1, 1], [1, 2]])
Y = np.array([[0], [0], [1], [1], [3]])
B = np.linalg.inv(dot(X.T, X)).dot(dot(X.T, Y))
print(B)
print("YHat = ", B[0, 0], " + ", B[1, 0], "x" )
```

In [ ]:
```
# Alternatively, use the NumPy stats linear regeretion function
x = np.array([-2, -1, 0, 1, 2])
y = np.array([0, 0, 1, 1, 3])
b1, b0, r, p_val, stderr = stats.linregress(x,y)
print("YBar = ", b0, " + ", b1, "x")
```